k-Chordal Graphs: from Cops and Robber to Compact Routing via Treewidth

Adrian Kosowski, Bi Li, Nicolas Nisse, Karol Suchan

k-Chordal Graphs: from Cops and Robber to Compact Routing via Treewidth[†]

Adrian Kosowski[1] and Bi Li[2,3] and Nicolas Nisse[2] and Karol Suchan [4]

[1] *CEPAGE, INRIA, LaBRI, Talence, France* [2]*MASCOTTE, INRIA, I3S(CNRS/UNS), Sophia Antipolis, France*
[3]*CAS & AAMS, Beijing, China* [4]*Univ. Adolfo Ibanez, Facultad de Ingenieria y Ciencias, Santiago, Chile*

Nous présentons un algorithme quadratique qui, étant donné un graphe G et un entier $k \geq 3$, certifie que G contient un cycle induit de longueur $> k$, ou calcule une décomposition arborescente de G dont chaque "sac" induit un k-caterpillar (graphe qui contient un chemin dominant, de longueur au plus $k-2$). Entre autre, ce résultat implique que les graphes k-cordaux (sans cycle induit de longueur $> k$) de maximum degree Δ ont une largeur arborescente $O(k \cdot \Delta)$, ce qui améliore la borne $\Delta(\Delta-1)^{k-3}$ de Bodlaender et Thilikos (1997). De plus, l'hyperbolicité d'un graphe admettant une telle décomposition est $\leq \lfloor \frac{3}{2}k \rfloor$. Pour tout graphe qui admet une telle décomposition, nous proposons un algorithme de routage compact utilisant des adresses, en-têtes et tables de routage de taille $O(k \log n)$ bits et de stretch $O(k \cdot \log \Delta)$. Au passage, nous montrons que $k-1$ policiers sont suffisants pour capturer un voleur dans un graphe k-cordal.

Keywords: Cordalité, hyperbolicité, décomposition arborescente, routage compact, un peu de gendarmes et voleur.

1 Introduction

Because of the huge size of actual networks, an important research effort is currently done to use their structural properties for algorithmic purpose. Indeed, in large-scale networks, algorithms that perform in polynomial time may become unpractical. Therefore, it is important to design algorithms depending only quadratically or linearly on the size of the network when its topology is expected to satisfy some properties. The *chordality* of a graph is the length of its longest induced (i.e., chordless) cycle. A *k-chordal graph* is a graph with chordality at most k. The *hyperbolicity* of a graph reflects how the metric (distances) of the graph is close to the metric of a tree (intuitively, in a graph with small hyperbolicity, any two shortest paths between the same pair of vertices are close to each other). Several recent works take advantage of both these properties for algorithmic issues (e.g., routing) in large-scale networks. Indeed, in such networks (e.g., the Internet), the high clustering coefficient implies that few large chordless cycles should exist, and their low diameter implies a small hyperbolicity [dMSV11].

In another context, *tree-decompositions* play an important role in the design of efficient algorithms. Simply speaking, a tree-decomposition is a mapping of a graph into a tree, where some connectivity constraints must be satisfied. The *width* of a tree-decomposition is the maximum size of its nodes (where the size of a node of the tree is the number of vertices of the graph mapped to this node) and the *treewidth* of a graph is the smallest width of its tree-decompositions. By using dynamic programming based on a tree-decomposition, many NP-hard problems have been shown to be linear time solvable for graph with bounded treewidth [CM93]. In particular, there are linear-time algorithms to compute an optimal tree-decomposition of a graph with bounded treewidth [Bod93]. However, on the practical point of view, this approach has several drawbacks. First, all above-mentioned algorithms are linear in the size of the graph but (at least) exponential in the treewidth. Moreover, due to the high clustering coefficient of large-scale networks, their treewidth is expected to be large [dMSV11]. To face these problems, we propose to focus on the structure of the nodes of the tree-decomposition, instead of trying to minimize their size.

One of the most important problems in actual networks concerns routing. Delivering a message from a sender to a receiver is very simple when all nodes can store the whole map of the network. Indeed, in such

[†]Les preuves peuvent être consultées ici : `http://hal.inria.fr/hal-00671861`

a case, each time the message reaches a new node, it can refer to the routing tables (where the information is stored locally) to decide which will be its next hop and to eventually reach its destination. In the context of large-scale networks, the map of the network is (generally) only partially known, and moreover such a method would require to store to many information in each node. The compact routing problem aims at computing a small amount of information to be stored in the routing tables such that any message will eventually reach its destination. Moreover, any message is expected to go quickly from its source s to its destination d, i.e., following a route of length at most $d(s,t)+\rho$ where $d(s,t)$ is the distance from s to t in the graph, and $\rho \geq 0$ is the *additive stretch* of the routing scheme. In general graphs, any routing scheme achieving shortest paths (i.e., with no stretch) requires routing tables with $\Omega(n\log n)$ bits per node [FG97]. Therefore, many works have considered special graphs' classes as *k-chordal graphs* [NSR, Dou05], graphs with bounded hyperbolicity [KPBV09, CDE$^+$12] or with particular decompositions (e.g., [AG06]). In trees, a routing scheme achieving shortest path with routing tables of size $O(\log n)$ bits is proposed in [FG01].

Last but not least, Cops and robber games aim at capturing a robber moving in a graph, using as few cops as possible. Numerous studies on these games allowed a better understanding of the graphs' structure [BN11]. The main result in this work is a structural characterization for a large family of graphs (including k-chordal graphs) which is somehow a side-effect of our study on cops and robber games in k-chordal graphs. This particular tree-decomposition, efficiently computable, has several algorithmic applications such as compact routing.

Our Contributions. Our main contribution is the design of an algorithm that, given a m-edge graph G with maximum degree Δ and an integer $k \geq 3$, either returns an induced cycle of length at least $k+1$ in G or compute a tree-decomposition of G with each bag of it has a dominating path of order $\leq k-1$. In the latter case, this ensures that G has treewidth at most $(k-1)(\Delta-1)+2$, tree-length at most k and hyperbolicity at most $\lfloor \frac{3}{2}k \rfloor$. Our algorithm performs in a greedy way in time $O(m^2)$. In particular, this shows that the treewidth of a k-chordal graph is upper bounded by $O(k \cdot \Delta)$ improving the exponential bound of [BT97]. The proposed algorithm is mainly derived from our proof of the fact that $k-1$ cops are sufficient to capture a robber in k-chordal graphs.Such a tree-decomposition may be used efficiently for solving problems using dynamic programming in graphs with small chordality and small degree. In this note, we focus on routing applications. We present a compact routing scheme that uses our tree-decomposition and that achieves a stretch $2k(\lceil \log \Delta \rceil + \frac{9}{4}) - 4$ with routing tables, addresses and message's header of size $O(k \log n)$.

Related Work. Chordality and hyperbolicity are both parameters measuring the tree-likeness of a graph. Some papers consider the relationship between both parameters [WZ11].In particular, the hyperbolicity of a k-chordal graph is at most k, but the gap may be arbitrary large (take a $3 \times n$-grid). Bodlaender and Thilikos proved that the treewidth of a k-chordal graph with maximum degree Δ is at most $\Delta(\Delta-1)^{k-3}$ [BT97]. While the hyperbolicity can be polynomially computable, no algorithm better than the $O(n^4)$-brute force algorithm is known to compute hyperbolicity of n-node graphs. The problem of computing the chordality of a graph G is NP-complete since it may be related to computing a longuest cycle in the graph obtained from G after sub-dividing all edges once. Finding the longest induced path is $W[2]$-complete [CF07] and the problem is Fixed Parameter Tractable in planar graphs [KK09]. Finally, k-chordal graphs have also been studied in the context of routing. The best known compact routing scheme in k-chordal graphs achieves a stretch of $k+1$ using routing tables of size $O(\log^2 n)$ bits [Dou05, CDE$^+$12].

2 A detour through Cops and Robber games

Given a graph G, a player starts by placing $k \geq 1$ cops on some vertices of G, then a visible robber is placed on one vertex of G. Alternatively, the cop-player may move each cop along one edge, and then the robber can move to an adjacent vertex. The robber is captured if, at some step, a cop occupies the same vertex. The *cop-number* $cn(G)$ of G is the least k such that k cops are sufficient to capture a robber in G whatever the robber does. A long standing conjecture due to Meyniel states that $cn(G) = O(\sqrt{n})$ for any n-node graph G [BN11]. To tackle this question, many researchers have focused on particular graph classes and provided many nice structural results [BN11]. We consider the class of k-chordal graphs. In the next section, we show how to adapt this simple strategy to compute a particular tree-decomposition.

Theorem 1 *Let $k \geq 3$. For any k-chordal graph G, $cn(G) \leq k-1$.*

Sketch of proof. The strategy is the following. First, all cops start at some arbitrary vertex. At some cop-turn, the cops occupy the vertices of chordless path $P = (v_1, \cdots, v_i)$ with $i \leq k-1$ vertices. If the robber occupies a vertex in $N[P]$ (a vertex of P or neighbor of P), then it is captured during the next move. Otherwise, it occupies a vertex of some connected component C of $G \setminus N[P]$. Roughly, if $i < k-1$, we prove that there is neighbor x of v_i such that $P' = P \cup \{x\}$ is a chordless path and $N(x) \cap C \neq \emptyset$. In that case, a cop goes to x (the cops occupy the vertices in P') and the robber is restricted to a connected component C' of $G \setminus N[P']$ with $C' \subset C$. The key argument is that, in k-chordal graphs, if $i = k-1$, then C is a connected component of $G \setminus N[P'']$ where $P'' = (v_2, \cdots, v_i)$. In this latter case, the cops on v_1 go to v_2 and the robber cannot leave C. Therefore, at each time, either the length of the path decreases without changing the area where the robber stands, or this area strictly decreases. Thus, the robber is eventually captured. □

3 Beautiful Tree-decomposition

In this section, we present our main contribution, that is, an algorithm that, given a m-edge graph G and an integer $k \geq 3$, either returns an induced cycle of length at least $k+1$ in G or compute a tree-decomposition of G with interesting properties. First, we need some definitions.

A *tree-decomposition* of a graph $G = (V,E)$ is a pair $(\{X_i | i \in I\}, T = (I,M))$, where T is a tree and $\{X_i | i \in I\}$ is a family of subsets, called bags, of vertices of G such that (1) $V = \cup_{i \in I} X_i$; (2) $\forall \{uv\} \in E$ there is $i \in I$ such that $u, v \in X_i$; and (3) $\forall v \in V$, $\{i \in I | v \in X_i\}$ induces a (connected) subtree of T. The *width* of a tree-decomposition is the size (minus 1) of its largest bag and its *ℓ-width* is the maximum diameter of the subgraphs induced by the bags. The *treewidth*, resp., *tree-length*, of a graph G is the minimum width, resp., ℓ-width, over all possible tree-decompositions of G.

Let $k \geq 2$. A k-caterpillar is a graph that has a dominating set which induces a chordless path of order at most $k-1$. A tree-decomposition is said *k-beautiful* if all its bags induce a k-caterpillar.

Theorem 2 *There is a $O(m^2)$-algorithm that takes a m-edge graph G and an integer $k \geq 3$ as inputs and :*

- *either returns an induced cycle of length at least $k+1$;*
- *or returns a k-beautiful tree-decomposition of G ;*

In particular, if the input graph is k-chordal, this algorithm always computes a k-beautiful tree-decomposition. It is obvious to see that, if a graph G with maximum degree Δ admits a k-beautiful tree-decomposition, then its treewidth is $\leq (k-1)(\Delta-1)+2$ and its tree-length is $\leq k$. We also prove that if a graph admits a k-beautiful tree-decomposition, then its hyperbolicity is $\leq \lfloor \frac{3}{2}k \rfloor$.

Corollary 1

- *Any k-chordal graph G with maximum degree Δ has treewidth at most $(k-1)(\Delta-1)+2$.*
- *There is an algorithm that, given a m-edge graph G and $k \geq 3$, states that either G has chordality at least $k+1$ or G has hyperbolicity at most $\lfloor \frac{3}{2}k \rfloor$, in time $O(m^2)$.*

Sketch of proof. The idea of the algorithm follows the strategy for the cops and robber game described above. Starting from any chordless path P of order $\leq k-1$ in G, we aim at computing a beautiful tree-decomposition of G with one bag being exactly $N[P]$. The easy case is when $N[P]$ separates the graph into several connected components C_i's. In such case, the algorithm simply proceeds independently on each subgraph $N[P] \cup C_i$ starting from P. By induction of the size of the subgraphs, either we obtain a beautiful tree-decomposition with one bag $N[P]$ for each subgraph and then, it is easy to glue them together, or the algorithm find a long induced cycle in one of the subgraphs.

If $N[P]$ does not separate the graph, then there are two cases. If $|V(P)| < k-1$, we show that P can be extended to a chordless path P' with one more vertex, and then the algorithm proceeds starting from P'. Otherwise (if $|V(P)| = k-1$), we show that either there is a chordless cycle with order $\geq k+1$ and containing P, or an end v of P can be "removed". More precisely, in the latter case, let P^* be the subpath of P after the removal of v, we show that $N[P^*] = N[P]$. Then the algorithm proceeds starting from P^*. □

4 Routing Scheme using k-beautiful decomposition

In this section, we consider the class of graphs that admit a k-beautiful tree-decomposition (this includes k-chordal graphs). For any n-node graph $G = (V,E)$ of this class and with maximum degree Δ, we combine the compact routing scheme in trees of [FG01] together with the k-beautiful tree-decomposition to obtain a routing scheme which uses message headers, addresses and local memories of size $O(k\log n)$ bits per node, and the additive stretch $2k(\lceil\log\Delta\rceil+\frac{9}{4})-4$. Due to lack of space, we only sketch the scheme.

Let us first describe the data structure we use. Let F be a BFS-tree of G rooted at r and let us assume that the vertices have received addresses and routing tables as in [FG01]. Therefore, in G, it is possible to route messages along the paths in F using addresses and routing tables of $O(\log n)$ bits per node. Moreover, let $(T=(I,M), X=\{X_i | i\in I\})$ be a k-beautiful tree-decomposition of G. Assume that T is rooted in $r'\in I$ with $r\in X_{r'}$, and that the nodes of I receive a label given by a DFS of T starting from r'. In that way, it is easy to know if $i\in I$ is an ancestor of $j\in I$ (this depends on the labels of i and j and on the size of the subtree rooted in i). For any vertex $v\in V$, let $i(v)$ be the label of the node $i\in I$ that is closest to r' in T and such that $v\in X_i$. The index $i(v)$ is added to the address of v and the full description (vertices and port numbers) of the dominating path (of order at most $k-1$) of the bag $X_{i(v)}$ is added to the routing table of v. This uses $O(k\log n)$ bits.

Now we are ready to describe the routing scheme. Roughly, the scheme consists in following the paths in tree F and using one bag of the decomposition as a short-cut between two branches of F. Intuitively, if the source s and the destination d are "far appart", then there is a bag X of the tree-decomposition that separates s and d in G. The message follows the path in F until it reaches X, then an exhaustive search is done in X until the message finds an ancestor y of d, and finally it follows the path from y to d in F.

More precisely, consider that a message must be sent from $s\in V$ to $d\in V$. If s is the ancestor of d in the BFS-tree F, then the message follows the path in F using the algorithm in [FG01] and the message reaches d via a shortest path in G. Otherwise, the message first follows the path from s to r in F until it reaches a vertex $x\in V$ such that $i(x)\in I$ is a common ancestor of $i(s)\in I$ and $i(d)\in I$ in T (possibly, $i(x)$ equals $i(s)$ or/and $i(d)$). The properties of tree-decomposition imply that there is $y\in X_{i(x)}$ that is an ancestor of d in F. First, the message copies the full description of the dominating path of $X_{i(x)}$ that appears in the routing table of x. Then using it and a particular labeling of the edges of G, the message finds y by doing a dichotomic search in $X_{i(x)}$. Once the message has reached y, it follows the path from y to d in F using the algorithm in [FG01]. We prove that the phases that follow the paths in F induce a stretch $O(k)$ and the dichotomic search induces a stretch $O(k\cdot\log\Delta)$.

Further Work. It would be interesting to reduce the $O(k\cdot\log\Delta)$ stretch due to the dichotomic search phase of our routing scheme. Another interesting topic concerns the computation of tree-decompositions not trying to minimize the size of the bag but imposing some specific algorithmically useful structure.

Références

[AG06] I. Abraham and C. Gavoille. Object location using path separators. In *PODC*, pages 188–197. ACM, 2006.

[BN11] A. Bonato and R. Nowakovski. *The game of Cops and Robber on Graphs*. American Math. Soc., 2011.

[Bod93] H. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. In *STOC*, pages 226–234, 93.

[BT97] H. L. Bodlaender and D. M. Thilikos. Treewidth for graphs with small chordality. *Disc. Ap. Maths*, 79(1-3) :45–61, 1997.

[CDE+12] Victor Chepoi, Feodor F. Dragan, Bertrand Estellon, Michel Habib, Yann Vaxès, and Yang Xiang. Additive spanners and distance and routing labeling schemes for hyperbolic graphs. *Algorithmica*, 62(3-4) :713–732, 2012.

[CF07] Y. Chen and J. Flum. On parameterized path and chordless path problems. In *CCC*, pages 250–263, 2007.

[CM93] B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *TCS*, 109 :49–82, 1993.

[dMSV11] F. de Montgolfier, M. Soto, and L. Viennot. Treewidth and hyperbolicity of the internet. In *10th IEEE International Symposium on Networking Computing and Applications (NCA)*, pages 25–32. IEEE Computer Society, 2011.

[Dou05] Y. Dourisboure. Compact routing schemes for generalised chordal graphs. *J. of Graph Alg. and App*, 9(2) :277–297, 2005.

[FG97] P. Fraigniaud and C. Gavoille. Universal routing schemes. *Distributed Computing*, 10(2) :65–78, 1997.

[FG01] P. Fraigniaud and C. Gavoille. Routing in trees. In *28th Int. Col. on Aut., Lang. and Prog. (ICALP)*, pages 757–772, 2001.

[KK09] Y. Kobayashi and K. Kawarabayashi. Algorithms for finding an induced cycle in planar graphs and bounded genus graphs. In *20th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1146–1155. SIAM, 2009.

[KPBV09] D.V. Krioukov, F. Papadopoulos, M. Boguñá, and A. Vahdat. Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces. *SIGMETRICS Performance Evaluation Review*, 37(2) :15–17, 2009.

[NSR] N. Nisse, K. Suchan, and I. Rapaport. Distributed computing of efficient routing schemes in generalized chordal graphs. *Theoretical Computer Science*. to appear.

[WZ11] Y. Wu and C. Zhang. Hyperbolicity and chordality of a graph. *Electr. J. Comb.*, 18(1), 2011.